EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	1	"6934947"	US-PGPUB; USPAT	ADJ	ON	2006/09/26 10:52
S2	679	(717/124).ccls. and (@ad<"20031020" or @prad<"20031020" or @rlad<"20031020")	US-PGPUB; USPAT	ADJ	ON	2006/10/02 07:33
S3	13	(embedded and (RTOS or (realtime OS))) and (717/124). ccls. and (@ad<"20031020" or @prad<"20031020" or @rlad<"20031020")	US-PGPUB; USPAT	ADJ	ON	2006/09/27 17:14
S4	1181	"2" and (embedded and (RTOS or (realtime OS)))	US-PGPUB; USPAT	ADJ	ON	2006/09/27 17:13
S5	679	(717/124).ccls. and (@ad<"20031020" or @prad<"20031020" or @rlad<"20031020")	US-PGPUB; USPAT	ADJ	ON	2006/09/27 17:14
S6	13	S5 and (embedded and (RTOS or (realtime OS)))	US-PGPUB; USPAT	ADJ	ON	2006/09/27 17:15
S7	5	S6 and schedul$3	US-PGPUB; USPAT	ADJ	ON	2006/09/27 17:16
S8	1018	(embedded and (RTOS or (realtime OS))) and (@ad<"20031020" or @prad<"20031020" or @rlad<"20031020")	US-PGPUB; USPAT	ADJ	ON	2006/09/28 10:08
S9	5	S8 and (source near2 code near2 generation)	US-PGPUB; USPAT	ADJ	ON	2006/09/27 17:27
S10	0	S8 and (real$1time operating system generation)	US-PGPUB; USPAT	ADJ	ON	2006/09/27 17:28
S11	0	S8 and ((real$1time near2 operating near2 system near$2) or (rtos)) generation	US-PGPUB; USPAT	ADJ	ON	2006/09/27 17:29
S12	1015	S8 and ((real$1time near2 operating near2 system near$2) or (rtos))	US-PGPUB; USPAT	ADJ	ON	2006/09/27 17:29
S13	1	S8 and ((real$1time near2 operating near2 system near$2) or (rtos)) near3 generation	US-PGPUB; USPAT	ADJ	ON	2006/09/27 17:29
S14	96	(embedded and (RTOS or (realtime OS))) and "717".clas. and (@ad<"20031020" or @prad<"20031020" or @rlad<"20031020")	US-PGPUB; USPAT	ADJ	ON	2006/09/28 10:10

EAST Search History

S15	25	S14 and (code near1 generation)	US-PGPUB; USPAT	ADJ	ON	2006/09/28 10:10
S16	1224	(717/106-109).ccls. and (@ad<"20031020" or @prad<"20031020" or @rlad<"20031020")	US-PGPUB; USPAT	ADJ	ON	2006/10/02 07:33
S17	250	S16 and ((real near3 time$3) or RTOS)	US-PGPUB; USPAT	ADJ	ON	2006/10/02 15:30
S18	94	S17 and ((task near2 management) or schedul$4 or multi$1task$3)	US-PGPUB; USPAT	ADJ	ON	2006/10/02 07:37
S19	48	S17 and ((task near2 management) or schedul$4 or multi$1task$3) and priority	US-PGPUB; USPAT	ADJ	ON	2006/10/02 07:38
S20	1224	(717/106-109).ccls. and (@ad<"20031020" or @prad<"20031020" or @rlad<"20031020")	US-PGPUB; USPAT	ADJ	ON	2006/10/04 10:41
S21	250	S20 and ((real near3 time$3) or RTOS)	US-PGPUB; USPAT	ADJ	ON	2006/10/02 14:46
S22	250	S21	US-PGPUB; USPAT	ADJ	ON	2006/10/02 14:46
S23	0	S21 and (synathesi$4 near3 source near1 code)	US-PGPUB; USPAT	ADJ	ON	2006/10/02 15:28
S24	0	S21 and (synathesi$4)	US-PGPUB; USPAT	ADJ	ON	2006/10/02 15:28
S25	0	S21 and (synthesi$4 near3 source near1 code)	US-PGPUB; USPAT	ADJ	ON	2006/10/02 15:29
S26	36	S21 and (synthesi$4)	US-PGPUB; USPAT	ADJ	ON	2006/10/02 15:29
S27	0	S21 and (synthesi$4 near2 source$2)	US-PGPUB; USPAT	ADJ	ON	2006/10/02 15:29
S28	250	S21	US-PGPUB; USPAT	ADJ	ON	2006/10/02 15:30
S29	7	S21 and (preemptive and priority and initial)	US-PGPUB; USPAT	ADJ	ON	2006/10/02 15:31
S30	1	"4130883".PN.	USPAT; USOCR	ADJ	ON	2006/10/02 16:24
S31	1224	(717/106-109).ccls. and (@ad<"20031020" or @prad<"20031020" or @rlad<"20031020")	US-PGPUB; USPAT	ADJ	ON	2006/10/11 14:03
S32	1224	S31	US-PGPUB; USPAT	ADJ	ON	2006/10/04 10:41

EAST Search History

S33	0	S31 and (timer based task)	US-PGPUB; USPAT	ADJ	ON	2006/10/04 10:41
S34	0	S31 and (timer near2 based near2 task)	US-PGPUB; USPAT	ADJ	ON	2006/10/04 10:42
S35	0	S31 and (periodical near2 task)	US-PGPUB; USPAT	ADJ	ON	2006/10/04 10:47
S36	3	S31 and (loop near2 task)	US-PGPUB; USPAT	ADJ	ON	2006/10/04 10:51
S37	0	("60493268").PN.	US-PGPUB; USPAT	OR	OFF	2006/10/04 10:51
S38	4	"60493268"	US-PGPUB; USPAT	ADJ	ON	2006/10/04 10:52
S39	7707	"20030807"	US-PGPUB; USPAT	ADJ	ON	2006/10/04 10:52
S40	0	("20030807").PN.	US-PGPUB; USPAT	OR	OFF	2006/10/04 10:52
S42	1	"6934947"	US-PGPUB; USPAT	ADJ	ON	2006/10/11 14:03
S43	1229	(717/106-109).ccls. and (@ad<"20031020" or @prad<"20031020" or @rlad<"20031020")	US-PGPUB; USPAT	ADJ	ON	2006/10/11 14:05
S44	1229	S43	US-PGPUB; USPAT	ADJ	ON	2006/10/11 14:03
S45	0	S43 and (preemptive near1 task)	US-PGPUB; USPAT	ADJ	ON	2006/10/11 14:04
S46	49	(preemptive near1 task)	US-PGPUB; USPAT	ADJ	ON	2006/10/11 14:04
S47	11	(preemptive task) and restore and (@ad<"20031020" or @prad<"20031020" or @rlad<"20031020")	US-PGPUB; USPAT	ADJ	ON	2006/10/11 14:08
S48	11	(preemptive task) and restore and (real time) and (@ad<"20031020" or @prad<"20031020" or @rlad<"20031020")	US-PGPUB; USPAT	ADJ	ON	2006/10/11 14:08

Results for "(desmet d.<in>au)" e-mail
Your search matched 9 of 1513808 documents.
A maximum of 100 results are displayed, 25 to a page, sorted by Relevance in Descending order.

» Search Options

View Session History

New Search

Modify Search

(desmet d.<in>au) Search

Check to search only within this results set

Display Format: Citation Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IET JNL	IET Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IET CNF	IET Conference Proceeding
IEEE STD	IEEE Standard

view selected items Select All Deselect All

1. **Magnetic and magneto-photoellipsometric evaluation of corrosion in met**
Parker, M.R.; Venkataram, S.; DeSmet, D.;
Magnetics, IEEE Transactions on
Volume 28, Issue 5, Part 2, Sep 1992 Page(s):2368 - 2370
Digital Object Identifier 10.1109/20.179494
AbstractPlus | Full Text: PDF(208 KB) **IEEE JNL**
Rights and Permissions

2. **Energy-aware runtime scheduling for embedded-multiprocessor SOCs**
Peng Yang; Chung Wong; Marchal, P.; Catthoor, F.; Desmet, D.; Verkest, D.; l
Design & Test of Computers, IEEE
Volume 18, Issue 5, Sept.-Oct. 2001 Page(s):46 - 58
Digital Object Identifier 10.1109/54.953271
AbstractPlus | References | Full Text: PDF(200 KB) **IEEE JNL**
Rights and Permissions

3. **Mapping concurrent applications on network-on-chip platforms**
Bartic, T.A.; Desmet, D.; Mignolet, J.-Y.; Miller, J.; Robert, F.;
Signal Processing Systems Design and Implementation, 2005. IEEE Workshop
2-4 Nov. 2005 Page(s):154 - 159
Digital Object Identifier 10.1109/SIPS.2005.1579856
AbstractPlus | Full Text: PDF(3715 KB) **IEEE CNF**
Rights and Permissions

4. **Operating system based software generation for systems-on-chip**
Desmet, D.; Verkest, D.; De Man, H.;
Design Automation Conference, 2000. Proceedings 2000. 37th
June 5-9, 2000 Page(s):396 - 401
AbstractPlus | Full Text: PDF(480 KB) **IEEE CNF**
Rights and Permissions

5. **Design of a secure, intelligent, and reconfigurable Web cam using a C ba design flow**
Verkest, D.; Desmet, D.; Avasare, P.; Coene, P.; Decneut, S.; Hendrickx, F.; M
Mignolet, J.-Y.; Pasko, R.; Schaumont, P.;
Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Conference on
Volume 1, 4-7 Nov. 2001 Page(s):463 - 467 vol.1
Digital Object Identifier 10.1109/ACSSC.2001.986969
AbstractPlus | Full Text: PDF(537 KB) **IEEE CNF**
Rights and Permissions

6. **Timed executable system specification of an ADSL modem using a C++ t environment: A case study**
Desmet, D.; Esvelt, M.; Avasare, P.; Verkest, D.; De Man, H.;

Hardware/Software Codesign, 1999. (CODES '99) Proceedings of the Seventh Workshop on
3-5 May 1999 Page(s):38 - 42
Digital Object Identifier 10.1109/HSC.1999.777388

AbstractPlus | Full Text: PDF(352 KB) **IEEE CNF**
Rights and Permissions

☐ **7. System design, optimization and intelligent code generation for standard processors**
Genin, D.; De Moortel, J.; Desmet, D.; Van de Velde, E.;
Circuits and Systems, 1989., IEEE International Symposium on
8-11 May 1989 Page(s):565 - 569 vol.1
Digital Object Identifier 10.1109/ISCAS.1989.100415

AbstractPlus | Full Text: PDF(320 KB) **IEEE CNF**
Rights and Permissions

☐ **8. Magnetic and magneto-photoellipsometric evaluation of corrosion in met**
Parker, M.R.; Venkataram, S.; DeSmet, D.;
Magnetics Conference, 1992. Digests of Intermag '92., International
13-16 April 1992 Page(s):113 - 113

AbstractPlus | Full Text: PDF(177 KB) **IEEE CNF**
Rights and Permissions

☐ **9. ASSYNT: efficient assembly code generation for digital signal processor: data flowgraph**
Desmet, D.; Genin, D.;
Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE Inte Conference on
Volume 3, 27-30 April 1993 Page(s):45 - 48 vol.3
Digital Object Identifier 10.1109/ICASSP.1993.319431

AbstractPlus | Full Text: PDF(320 KB) **IEEE CNF**
Rights and Permissions

Terms used **embedded rtos source code generation** Found **105,379** of **198,310**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 System design methods: scheduling advances: Extended quasi-static scheduling for formal synthesis and code generation of embedded software

Feng-Shi Su, Pao-Ann Hsiung

May 2002 **Proceedings of the tenth international symposium on Hardware/software codesign CODES '02**

Publisher: ACM Press

Full text available: pdf(590.26 KB) Additional Information: full citation, abstract, references, citings, index terms

With the computerization of most daily-life amenities such as home appliances, the software in a real-time embedded system now accounts for as much as 70% of a system design. On one hand, this increase in software has made embedded systems more accessible and easy to use, while on the other hand, it has also necessitated further research on how complex embedded software can be designed automatically and correctly. Enhancing recent advances in this research, we propose an *Extended Quasi-Static ...*

2 The embedded machine: predictable, portable real-time code

Thomas A. Henzinger, Christoph M. Kirsch

May 2002 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation PLDI '02**, Volume 37 Issue 5

Publisher: ACM Press

Full text available: pdf(223.85 KB) Additional Information: full citation, abstract, references, citings, index terms

The Embedded Machine is a virtual machine that mediates in real time the interaction between software processes and physical processes. It separates the compilation of embedded programs into two phases. The first, platform-independent compiler phase generates E code (code executed by the Embedded Machine), which supervises the timing ---not the scheduling--- of application tasks relative to external events, such as clock ticks and sensor interrupts. E~code is portable and exhibits, given an inpu ...

Keywords: real time, virtual machine

3 Advances in hardware/software co-simulation techniques: RTOS-centric hardware/software cosimulator for embedded system design

Shinya Honda, Takayuki Wakabayashi, Hiroyuki Tomiyama, Hiroaki Takada

September 2004 **Proceedings of the 2nd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis CODES+ISSS '04**

Publisher: ACM Press

Full text available: pdf(510.21 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents an RTOS-centric hardware/software cosimulator which we have

developed for embedded system design. One of the most remarkable features in our cosimulator is that it has a complete simulation model of an RTOS which is widely used in industry, so that application tasks including RTOS service calls are natively executed on a host computer. Our cosimulator also features cosimulation with functional simulation models of hardware written in C/C++ and cosimulation with HDL simulators ...

Keywords: RTOS, cosimulation, embedded Systems

4 Synthesis of time-constrained multitasking embedded software

André C. Nácul, Tony Givargis

October 2006 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 11 Issue 4

Publisher: ACM Press

Full text available: pdf(1.50 MB) Additional Information: full citation, abstract, references, index terms

In modern embedded systems, software development plays a vital role. Many key functions are being migrated to software, aiming at a shorter time to market and easier upgrades. Multitasking is increasingly common in embedded software, and many of these tasks incorporate real-time constraints. Although multitasking simplifies coding, it demands an operating system and imposes significant overhead on the system. The use of serializing compilers, such as the Phantom compiler, allows the synthesis of ...

Keywords: Code serialization, multitasking, real-time embedded software, software synthesis

5 Application Mapping to a Hardware Platform through Automated Code Generation Targeting a RTOS: A Design Case Study

Monica Besana, Michele Borgatti

March 2003 **Proceedings of the conference on Design, Automation and Test in Europe: Designers' Forum - Volume 2 DATE '03**

Publisher: IEEE Computer Society

Full text available: pdf(493.15 KB) Publisher Site Additional Information: full citation, abstract, citings, index terms

Consistency,accuracy and efficiency are key aspects for practical sability of a system design flow featuring automatic code generation.Consistency is the property of maintaining the same behavio r at different levels of abstraction through synthesis and refinement,leading to functionally correct implementation.Accuracy is the property of having a good estimation of system performances while evaluating a high-level representation of the system.Efficiency is the property of introducing low overhea ...

6 Scheduling for embedded systems: POSIX modeling in SystemC

Hector Posadas, Jesús Ádamez, Pablo Sánchez, Eugenio Villar, Francisco Blasco

January 2006 **Proceedings of the 2006 conference on Asia South Pacific design automation ASP-DAC '06**

Publisher: ACM Press

Full text available: pdf(252.01 KB) Additional Information: full citation, abstract, references

Early estimation of the execution time of Real-Time embedded SW is an essential task in complex, HW/SW embedded system design. Application SW execution time estimation requires taking into account the impact of the underlying RTOS. As a consequence, RTOS modeling is becoming an active research area. SystemC provides a framework for multiprocessing, HW/SW co-simulation at several abstraction levels. In this paper, a SystemC library for POSIX modeling and simulation is presented. By using the libr ...

7 System level modeling and embedded software: Embedded software generation from system level specification for multi-tasking embedded systems

KiSeun Kwon, YoungMin Yi, DoHyung Kim, SoonHoi Ha

January 2005 **Proceedings of the 2005 conference on Asia South Pacific design automation ASP-DAC '05**

Publisher: ACM Press

Full text available: pdf(230.51 KB) Additional Information: full citation, abstract, references

In this paper we present a new design flow in which embedded software code is generated from system level specification of multi-tasking embedded system, both for simulation and implementation. The generated software has a layered structure using virtual OS APIs and OS wrapper implementations to make it reconfigurable for multiple target platforms. Implementation of the OS wrapper is explained in details. With a Divx play example, we show some experimental results about the real-time performance ...

8 Operating system based software generation for systems-on-chip

Dirk Desmet, D. Verkest, Hugo De Man

June 2000 **Proceedings of the 37th conference on Design automation DAC '00**

Publisher: ACM Press

Full text available: pdf(65.66 KB) Additional Information: full citation, abstract, references, citings, index terms

In this paper we propose a system-level design environment, aimed at System-on-Chip (SOC) designs, including real-time embedded software. While many SOC modeling languages originate from hardware description languages, and thus tend to describe statical architectures, we observe that embedded software makes SOC designs essentially dynamic, and so a SOC modeling environment must include dynamic behavior. Such behavior is analogous to the services an Operating System offers in the software wo ...

9 Porting RTOS device drivers to embedded Linux

Bill Weinberg

October 2004 **Linux Journal**, Volume 2004 Issue 126

Publisher: Specialized Systems Consultants, Inc.

Full text available: html(22.60 KB) Additional Information: full citation, abstract

Your old real-time operating system made you do a lot for yourself as a driver author. Take advantage of the facilities Linux offers and clean up some spaghetti code while you're at it.

10 Software thread integration for embedded system display applications

Alexander G. Dean

February 2006 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 5 Issue 1

Publisher: ACM Press

Full text available: pdf(1.40 MB) Additional Information: full citation, abstract, references, index terms

Embedded systems require control of many concurrent real-time activities, leading to system designs that feature a variety of hardware peripherals, with each providing a specific, dedicated service. These peripherals increase system size, cost, weight, and design time. Software thread integration (STI) provides low-cost thread concurrency on general-purpose processors by automatically interleaving multiple threads of control into one. This simplifies hardware to software migration (which elimina ...

Keywords: fine-grain concurrency, hardware to software migration, software thread integration

11 UML-based multiprocessor SoC design framework

Tero Kangas, Petri Kukkala, Heikki Orsila, Erno Salminen, Marko Hännikäinen, Timo D. Hämäläinen, Jouni Riihimäki, Kimmo Kuusilinna

May 2006 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 5 Issue 2

Publisher: ACM Press

Full text available: pdf(3.37 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes a complete design flow for multiprocessor systems-on-chips (SoCs) covering the design phases from system-level modeling to FPGA prototyping. The design of complex heterogeneous systems is enabled by raising the abstraction level and providing several system-level design automation tools. The system is modeled in a UML design environment following a new UML profile that specifies the practices for orthogonal application and architecture modeling. The design flow tools are gov ...

Keywords: UML 2.0, architecture exploration, design flow

12 A customizable library to support software synthesis for embedded applications and micro-kernel systems

Carsten Ditze

September 1998 **Proceedings of the 8th ACM SIGOPS European workshop on Support for composing distributed applications EW 8**

Publisher: ACM Press

Full text available: pdf(1.17 MB) Additional Information: full citation, index terms

13 Scheduling refinement in abstract RTOS models

Fabiano Hessel, Vitor M. Da Rosa, Carlos Eduardo Reif, César Marcon, Tatiana Gadelha Serra Dos Santos

May 2006 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 5 Issue 2

Publisher: ACM Press

Full text available: pdf(400.69 KB) Additional Information: full citation, abstract, references, index terms

Scheduling decision for real-time embedded software applications has a great impact on system performance and, therefore, is an important issue in RTOS design. Moreover, it is highly desirable to have the system designer able to evaluate and select the right scheduling policy at high abstraction levels, in order to allow faster exploration of the design space. In this paper, we address this problem by introducing an abstract RTOS model, as well as a new approach to refine an unscheduled high-lev ...

Keywords: RTOS scheduling, Real-time operating systems, transaction level Modeling

14 Lightweight Multitasking Support for Embedded Systems using the Phantom Serializing Compiler

Andre C. Nacul, Tony Givargis

March 2005 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 2 DATE '05**

Publisher: IEEE Computer Society

Full text available: pdf(153.76 KB) Additional Information: full citation, abstract, citings, index terms

Embedded software continues to play an ever increasing role in the design of complex embedded applications. In part, the elevated level of abstraction provided by a high-level programming paradigm immensely facilitates a short design cycle, fewer errors, portability, and reuse. Serializing compilers have been proposed as an alternative to traditional OS techniques, enabling a designer to develop multitasking applications without the need of OS support. In this work, we outline the inner workings ...

15 Power analysis of embedded operating systems

Robert P. Dick, Ganesh Lakshminarayana, Anand Raghunathan, Niraj K. Jha

June 2000 **Proceedings of the 37th conference on Design automation DAC '00**

Publisher: ACM Press

Full text available: pdf(225.31 KB) Additional Information: full citation, abstract, references, citings, index terms

The increasing complexity and software content of embedded systems has led to the frequent use of system software that helps applications access underlying hardware resources easily and efficiently. In this paper, we analyze the power consumption of real-time operating systems (RTOSs), which form an important component of the system software layer. Despite the widespread use of, and significant role played by, RTOSs in mobile and low-power embedded systems, little is known about their power ...

16 Automatic generation of scheduling and communication code in real-time parallel programs

André Bakkers, Johan Sunter, Evert Ploeg

November 1995 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1995 workshop on Languages, compilers, & tools for real-time systems LCTES '95**, Volume 30 Issue 11

Publisher: ACM Press

Full text available: pdf(1.45 MB) Additional Information: full citation, abstract, references, index terms

Inter-process communication and scheduling are notorious problem areas in the design of real-time systems. Using CASE tools, the system design phase will in general result in a system description in the form of parallel processes. Manual allocation of these processes to processors may result in error prone and/or slow communication code. Scheduling of the processes, necessary to meet timing constraints, is also a tedious task that takes many iterations. The described design tools result in code ...

17 Embedded application design using a real-time OS

David Stepner, Nagarajan Rajan, David Hui

June 1999 **Proceedings of the 36th ACM/IEEE conference on Design automation DAC '99**

Publisher: ACM Press

Full text available: pdf(105.02 KB) Additional Information: full citation, references, citings, index terms

18 System-level power optimization: techniques and tools

Luca Benini, Giovanni de Micheli

April 2000 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 5 Issue 2

Publisher: ACM Press

Full text available: pdf(385.22 KB) Additional Information: full citation, abstract, references, citings, index terms

This tutorial surveys design methods for energy-efficient system-level design. We consider electronic sytems consisting of a hardware platform and software layers. We consider the three major constituents of hardware that consume energy, namely computation, communication, and storage units, and we review methods of reducing their energy consumption. We also study models for analyzing the energy cost of software, and methods for energy-efficient software design and compilation. This survery ...

19 Open Source Software for Real-time Solutions

Charles Curley

October 1999 **Linux Journal**

Publisher: Specialized Systems Consultants, Inc.

Full text available: html(17.99 KB) Additional Information: full citation, abstract, references, index terms

Mr. Curley takes a look at two open-source solutions for embedded systems: RTLinux and eCOS from Cygnus

20 OOPSLA practitioner reports chair's welcome: OO techniques applied to a real-time, embedded, spaceborne application

Alexander T. Murray, Mohammad Shahabuddin

October 2006 **Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06**

Publisher: ACM Press

Full text available: pdf(510.33 KB) Additional Information: full citation, abstract, references

Though Object-Oriented Analysis, Design, and languages have become the dominant practices in many, or most, domains of software engineering, concerns about complexity, size, and performance in the embedded, real-time software domain have led to a prevalent view that OO technology is not suitable for the domain. We challenge this view through a successful application of OOA, OOD, and C++ (including STL) in the embedded, real-time flight software in an Earth-orbiting science instrument named Aquar ...

Keywords: C++, embedded, object-oriented analysis, object-oriented design, real-time, unified modeling language, use case

Terms used **embedded rtos source code generation** Found **105,379** of **198,310**

Sort results by: relevance — Save results to a Binder — Search Tips — Open results in a new window

Display results: expanded form

Try an Advanced Search
Try this search in The ACM Guide

Results 21 - 40 of 200 Result page: previous 1 **2** 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

21 Embedded systems: hardware/software design methodology and optimization: A comparison of the RTU hardware RTOS with a hardware/software RTOS

Jaehwan Lee, Vincent John Mooney, Anders Daleby, Karl Ingström, Tommy Klevin, Lennart Lindh

January 2003 **Proceedings of the 2003 conference on Asia South Pacific design automation ASPDAC**

Publisher: ACM Press

Full text available: pdf(237.77 KB) Additional Information: full citation, abstract, references, citings

In this paper, we show the performance comparison and analysis result among three RTOSes: the Real-Time Unit (RTU) hardware RTOS, the pure software Atalanta RTOS and a hardware/software RTOS composed of part of Atalanta interfaced to the System-on-a-Chip Lock Cache (SoCLC) hardware. We also present our RTOS configuration framework that can automatically configure these three RTOSes. The average-case simulation result of a database application example on a three-processor system running thirty ta ...

22 Synthesis of embedded software using free-choice Petri nets

Marco Sgroi, Luciano Lavagno, Yosinori Watanabe, Alberto Sangiovanni-Vincentelli

June 1999 **Proceedings of the 36th ACM/IEEE conference on Design automation DAC '99**

Publisher: ACM Press

Full text available: pdf(121.92 KB) Additional Information: full citation, references, citings, index terms

23 Platform Based on Open-Source Cores for Industrial Applications

M. Bolado, H. Posadas, J. Castillo, P. Huerta, P. Sánchez, C. Sánchez, H. Fouren, F. Blasco

February 2004 **Proceedings of the conference on Design, automation and test in Europe - Volume 2 DATE '04**

Publisher: IEEE Computer Society

Full text available: pdf(94.33 KB) Additional Information: full citation, abstract, citings, index terms

The latest version of the International Technology Roadmap for Semiconductors predicts that design reuse will be essential in the near future to face the constantly increasing design complexity. The concept comes from software engineering in which reuse is a fundamentaltechnology. In order to provide libraries and applications to reuse in software development, some open-source initiatives (e.g. Linux, gcc, X, mysql) have appeared during the last decades. The basic idea is to distribute the libra ...

24 EMERALDS: a small-memory real-time microkernel

Khawar M. Zuberi, Padmanabhan Pillai, Kang G. Shin

December 1999 **ACM SIGOPS Operating Systems Review , Proceedings of the seventeenth ACM symposium on Operating systems principles SOSP '99**, Volume 33 Issue 5

Publisher: ACM Press

Full text available: pdf(1.59 MB) Additional Information: full citation, abstract, references, citings, index terms

EMERALDS (Extensible Microkernel for Embedded, ReAL-time, Distributed Systems) is a real-time microkernel designed for small-memory embedded applications. These applications must run on slow (15-25MHz) processors with just 32-128 kbytes of memory, either to keep production costs down in mass-produced systems or to keep weight and power consumption low. To be feasible for such applications, the OS must not only be small in size (less than 20 kbytes), but also have low-overhead kernel services. Un ...

25 An efficient and lightweight embedded Web server for Web-based network element management

Hong-Taek Ju, Mi-Joung Choi, James W. Hong

September 2000 **International Journal of Network Management**, Volume 10 Issue 5

Publisher: John Wiley & Sons, Inc.

Full text available: pdf(428.26 KB) Additional Information: full citation, abstract, references, index terms

An Embedded Web Server (EWS) is a Web server which runs on an embedded system with limited computing resources to serve embedded Web documents to a Web browser. By embedding a Web server into a network device, it is possible to provide a Web‐based management user interface, which are user‐friendly, inexpensive, cross‐platform, and network‐ready. This article explores the topic of an efficient and lightweight embedded Web server for Web‐based netw ...

26 Software testing: Random testing of interrupt-driven software

John Regehr

September 2005 **Proceedings of the 5th ACM international conference on Embedded software EMSOFT '05**

Publisher: ACM Press

Full text available: pdf(387.96 KB) Additional Information: full citation, abstract, references, index terms

Interrupt-driven embedded software is hard to thoroughly test since it usually contains a very large number of executable paths. Developers can test more of these paths using *random interrupt testing*---firing random interrupt handlers at random times. Unfortunately, naïve application of random testing to interrupt-driven software does not work: some randomly generated interrupt schedules violate system semantics, causing spurious failures. The contribution of this paper is the design ...

Keywords: embedded systems, interrupt-driven software, random testing, sensor networks

27 Converging CSP specifications and C++ programming via selective formalism

William B. Gardner

May 2005 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 4 Issue 2

Publisher: ACM Press

Full text available: pdf(617.07 KB) Additional Information: full citation, abstract, references, citings, index terms

CSP (communicating sequential processes) is a useful algebraic notation for creating a hierarchical behavioral specification for concurrent systems, due to its formal interprocess synchronization and communication semantics. CSP specifications are amenable to simulation and formal verification by model-checking tools. A translator has been created to synthesize C++ code from CSP for execution with an object-oriented framework called CSP++, thereby making CSP specifications di ...

Keywords: Executable specifications, hardware/software codesign, object-oriented application frameworks

28 Embedded Control Systems Development with Giotto

Thomas A. Henzinger, Benjamin Horowitz, Christoph Meyer Kirsch

August 2001 **ACM SIGPLAN Notices , Proceedings of the 2001 ACM SIGPLAN**

workshop on Optimization of middleware and distributed systems OM '01 , Proceedings of the ACM SIGPLAN workshop on Languages, compilers and tools for embedded systems LCTES '01, Volume 36 Issue 8
Publisher: ACM Press

Full text available: pdf(280.43 KB) Additional Information: full citation, abstract, references, citings, index terms

Giotto is a principled, tool-supported design methodology for implementing embedded control systems on platforms of possibly distributed sensors, actuators, CPUs, and networks. Giotto is based on the principle that time-triggered task invocations plus time-triggered mode switches can form the abstract essence of programming real-time control systems. Giotto consists of a programming language with a formal semantics, and a retargetable compiler and runtime library. Giotto supports the automati ...

29 Departments: Interview: A Conversation with Jim Ready
Randy Harr
April 2003 **Queue**, Volume 1 Issue 2
Publisher: ACM Press
Full text available: pdf(160.55 KB) html(39.72 KB) Additional Information: full citation, citings, index terms

30 Session 52: high-performance simulation of transaction level and dataflow models: A model-driven design environment for embedded systems
E. Riccobene, P. Scandurra, A. Rosti, S. Bocchio
July 2006 **Proceedings of the 43rd annual conference on Design automation DAC '06**
Publisher: ACM Press
Full text available: pdf(671.00 KB) Additional Information: full citation, abstract, references, index terms

This paper presents a prototype environment for HW/SW co--design of embedded systems based on the Unified Modeling Language (UML) and SystemC. The environment supports a model-driven SoC design methodology which provides a graphical high-level representation of hardware and software components, and allows either C/C++/SystemC code generation from models and a reverse engineering process from code to graphical UML models.

Keywords: HW/SW co-design, MDA, SystemC, UML

31 Preventing interrupt overload
John Regehr, Usit Duongsaa
June 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools for embedded systems LCTES '05**, Volume 40 Issue 7
Publisher: ACM Press

Full text available: pdf(291.66 KB) Additional Information: full citation, abstract, references, citings, index terms

Performance guarantees can be given to tasks in an embedded system by ensuring that access to each shared resource is mediated by an appropriate scheduler. However, almost all previous work on CPU scheduling has focused on thread-level scheduling, resulting in systems that are vulnerable to a lower-level form of overload that occurs when too many interrupts arrive. This paper describes three new techniques, two software-based and one hardware-based, for creating systems that delay or drop excess ...

Keywords: embedded, interrupts, overload, scheduling

32 System-Level Performance Analysis in SystemC
H. Posadas, F. Herrera, P. Sánchez, E. Villar, F. Blasco
February 2004 **Proceedings of the conference on Design, automation and test in Europe - Volume 1 DATE '04**
Publisher: IEEE Computer Society

Full text available: pdf(123.80 KB) Additional Information: full citation, abstract, citings, index terms

As both the ITRS and the Medea+ DA Roadmaps have highlighted, early performance estimation is an essential step in any SoC design methodology [1-2]. This paper presents a C++ library for timing estimation at system level. The library is based on a general and systematic methodology that takes as input the original SystemC source code without anymodification and provides the estimation parameters by simply including the library within a usual simulation. As a consequence, the same models of compu ...

33 A decade of reconfigurable computing: a visionary retrospective

R. Hartenstein

March 2001 **Proceedings of the conference on Design, automation and test in Europe DATE '01**

Publisher: IEEE Press

Full text available: pdf(768.00 KB) Additional Information: full citation, references, citings, index terms

34 Don't care-based BDD minimization for embedded software

Youpyo Hong, Peter A. Beerel, Luciano Lavagno, Ellen M. Sentovich

May 1998 **Proceedings of the 35th annual conference on Design automation DAC '98**

Publisher: ACM Press

Full text available: pdf(167.37 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper explores the use of don't cares in software synthesis for embedded systems. Embedded systems have extremely tight real-time and code/data size constraints, that make expensive optimizations desirable. We propose applying BDD minimization techniques in the presence of a don't care set to synthesize code for extended Finite State Machines from a BDD-based representation of the FSM transition function. The don't care set can be derived from local analysis (such as unused sta ...

Keywords: MPEG4, codec, design automatian, flip-flops, level converters, low power, placement, synthesis, voltage scaling

35 MANTIS OS: an embedded multithreaded operating system for wireless micro sensor platforms

Shah Bhatti, James Carlson, Hui Dai, Jing Deng, Jeff Rose, Anmol Sheth, Brian Shucker, Charles Gruenwald, Adam Torgerson, Richard Han

August 2005 **Mobile Networks and Applications**, Volume 10 Issue 4

Publisher: Kluwer Academic Publishers

Full text available: pdf(1.27 MB) Additional Information: full citation, abstract, references, index terms

The MANTIS MultimodAl system for NeTworks of In-situ wireless Sensors provides a new multithreaded cross-platform embedded operating system for wireless sensor networks. As sensor networks accommodate increasingly complex tasks such as compression/aggregation and signal processing, preemptive multithreading in the MANTIS sensor OS (MOS) enables micro sensor nodes to natively interleave complex tasks with time-sensitive tasks, thereby mitigating the bounded buffer producer-consumer problem. To ac ...

Keywords: cross-platform, dynamic reprogramming, embedded operating system, lightweight, low power, multithreaded, sensor networks

36 An evaluation of the VME architecture for use in embedded systems education

Kenneth G. Ricks, David J. Jackson, William A. Stapleton

October 2005 **ACM SIGBED Review**, Volume 2 Issue 4

Publisher: ACM Press

Full text available: pdf(546.82 KB) Additional Information: full citation, abstract, references, index terms

The VMEbus is an IEEE standard architecture upon which many embedded and real-time systems are built. It has existed for nearly 25 years and has been extensively used for

military, industrial, and aerospace applications. This paper describes the general characteristics of the VMEbus architecture, specifically relating these characteristics to aspects of embedded systems education included as components of the IEEE/ACM CE2004 computer engineering model curriculum. Portions of this model curriculu ...

Keywords: computer architecture, computer engineering education, educational technology, embedded systems

37 Using Linux in Embedded and Real-Time Systems

Rick Lehrbaum
July 2000 **Linux Journal**
Publisher: Specialized Systems Consultants, Inc.
Full text available: html(13.87 KB) Additional Information: full citation, abstract, references, index terms

When you need an embedded operating system, Linux is a good place to start. Here's why.

38 Operating systems: Protothreads: simplifying event-driven programming of memory-constrained embedded systems

Adam Dunkels, Oliver Schmidt, Thiemo Voigt, Muneeb Ali
October 2006 **Proceedings of the 4th international conference on Embedded networked sensor systems SenSys '06**
Publisher: ACM Press
Full text available: pdf(307.33 KB) Additional Information: full citation, abstract, references, index terms

Event-driven programming is a popular model for writing programs for tiny embedded systems and sensor network nodes. While event-driven programming can keep the memory overhead down, it enforces a state machine programming style which makes many programs difficult to write, maintain, and debug. We present a novel programming abstraction called protothreads that makes it possible to write event-driven programs in a thread-like style, with a memory overhead of only two bytes per protothread. We sh ...

Keywords: embedded systems, threads, wireless sensor networks

39 Software Streaming via Block Streaming

Pramote Kuacharoen, Vincent J. Mooney, Vijay K. Madisetti
March 2003 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 1 DATE '03**
Publisher: IEEE Computer Society
Full text available: pdf(168.48 KB) Publisher Site Additional Information: full citation, abstract, index terms

Software streaming allows the execution of stream-enabled software on a device even while the transmission/streaming may still be in progress. Thus, the software can be executed while it is being streamed instead of causing the user to wait for the completion of download, decompression, installation and reconfiguration. Our streaming method can reduce application load time seen by the user since the application can start running as soon as the first executable unit is loaded into the memory. Fur ...

40 An Interview with Inder Singh

Jason Schumaker, Don Marti
September 2000 **Linux Journal**
Publisher: Specialized Systems Consultants, Inc.
Full text available: html(23.74 KB) Additional Information: full citation, index terms

Terms used **embedded rtos source code generation** Found **105,379** of **198,310**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 41 - 60 of 200 Result page: previous 1 2 **3** 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

41 Support for real time and OS services in embedded systems: Synthesizing operating system based device drivers in embedded systems

Shaojie Wang, Sharad Malik

October 2003 **Proceedings of the 1st IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis CODES+ISSS '03**

Publisher: ACM Press

Full text available: pdf(205.33 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents a correct-by-construction synthesis method for generating operating system based device drivers from a formally specified device behavior model. Existing driver development is largely manual using an ad-hoc design methodology. Consequently, this task is error prone and becomes a bottleneck in embedded system design methodology.Our solution to this problem starts by accurately specifying device access behavior with a formal model, viz. extended event driven finite state machin ...

Keywords: correct-by-construction, device driver, embedded system software, operating system based software synthesis

42 Reliability and security: Memory overflow protection for embedded systems using run-time checks, reuse and compression

Surupa Biswas, Matthew Simpson, Rajeev Barua

September 2004 **Proceedings of the 2004 international conference on Compilers, architecture, and synthesis for embedded systems CASES '04**

Publisher: ACM Press

Full text available: pdf(253.51 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Out-of-memory errors are a serious source of unreliability in most embedded systems. Applications run out of main memory because of the frequent difficulty of estimating the memory requirement before deployment, either because it depends on input data, or because certain language features prevent estimation. The typical lack of disks and virtual memory in embedded systems has two serious consequences when an out-of-memory error occurs. First, there is no swap space for the application to grow in ...

Keywords: data compression, heap overflow, out-of-memory errors, reliability, reuse, runtime checks, stack overflow

43 Real time analysis and priority scheduler generation for hardware-software systems with a synthesized run-time system

Vincent J. Mooney, Giovanni De Micheli

November 1997 **Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design ICCAD '97**

Publisher: IEEE Computer Society

Full text available: pdf(283.64 KB) Additional Information: full citation, abstract, references, citings, index terms
Publisher Site

We present a tool, called Clara, that performs real-time analysis and priority assignment for software tasks in a mixed hardware-software system with a custom run-time scheduler. We start from a system described in tasks/threads consisting of hardware specified in Verilog and software specified in C. We obtain the worst case execution time for each individual task. Then, based on the control flow of the application, Clara uses a dynamic programmming algorithm to automatically find optimal priori ...

Keywords: hardware-software codesign, real-time analysis, run-time scheduler, worst-case execution time, rtos

44 Guidelines for a graduate curriculum on embedded software and systems

P. Caspi, A. Sangiovanni-Vincentelli, L. Almeida, A. Benveniste, B. Bouyssounouse, G. Buttazzo, I. Crnkovic, W. Damm, J. Engblom, G. Folher, M. Garcia-Valls, H. Kopetz, Y. Lakhnech, F. Laroussinie, L. Lavagno, G. Lipari, F. Maraninchi, Ph. Peti, J. de la Puente, N. Scaife, J. Sifakis, R. de Simone, M. Torngren, P. Veríssimo, A. J. Wellings, R. Wilhelm, T. Willemse, W. Yi

August 2005 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 4 Issue 3

Publisher: ACM Press

Full text available: pdf(143.68 KB) Additional Information: full citation, abstract, citings, index terms

The design of embedded real-time systems requires skills from multiple specific disciplines, including, but not limited to, control, computer science, and electronics. This often involves experts from differing backgrounds, who do not recognize that they address similar, if not identical, issues from complementary angles. Design methodologies are lacking in rigor and discipline so that demonstrating correctness of an embedded design, if at all possible, is a very expensive proposition that may d ...

Keywords: Graduate curriculum, architecture and design, control, distributed systems, embedded systems, extrafunctional properties, labs, real-time

45 Meeting the Embedded Design Needs of Automotive Applications

Wayne Lyons

March 2005 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 3 DATE '05**

Publisher: IEEE Computer Society

Full text available: pdf(205.16 KB) Additional Information: full citation, abstract, citings, index terms

The importance of embedded systems in driving innovation in automotive applications continues to grow. Understanding the specific needs of developers targeting this market is also helping to drive innovation in RISC core design. This paper describes how a RISC instruction set architecture has evolved to better meet those needs, and the key implementation features in two very different RISC cores are used to demonstrate the challenges of designing for real-time automotive systems.

46 Embedded software: Automatic scenario detection for improved WCET estimation

Stefan Valentin Gheorghita, Sander Stuijk, Twan Basten, Henk Corporaal

June 2005 **Proceedings of the 42nd annual conference on Design automation DAC '05**

Publisher: ACM Press

Full text available: pdf(942.59 KB) Additional Information: full citation, abstract, references, citings, index terms

Modern embedded applications usually have real-time constraints and they are implemented using heterogeneous multiprocessor systems-on-chip. Dimensioning a system requires accurate estimations of the worst-case execution time (WCET). Overestimation leads to over-dimensioning. This paper introduces a method for automatic discovery of scenarios that incorporate correlations between different parts of applications. It is based on the application parameters with a large impact on the execution time. ...

Keywords: WCET, real-time, scenarios

47 Software portability gains realized with METAH and Ada95

Bruce Lewis
April 2002 **ACM SIGAda Ada Letters , Proceedings of the 11th international workshop on Real-time Ada workshop IRTAW '02**, Volume XXII Issue 4
Publisher: ACM Press
Full text available: pdf(235.40 KB) Additional Information: full citation, abstract, references

MetaH is an Architecture Description Language (ADL) developed to express and evaluate the software architecture of avionics and flight control systems. It is intended for not only description and analysis, but also for integration of the software components on the specified embedded hardware. This automated composition to specification with glue code generation allows rapid development and evolution of real-time embedded mission and safety critical systems. It also provides very high portability ...

48 Software performance estimation strategies in a system-level design tool

Jwahar R. Bammi, Wido Kruijtzer, Luciano Lavagno, Edwin Harcourt, Mihai T. Lazarescu
May 2000 **Proceedings of the eighth international workshop on Hardware/software codesign CODES '00**
Publisher: ACM Press
Full text available: pdf(187.67 KB) Additional Information: full citation, abstract, references, citings, index terms

High-level cost and performance estimation, coupled with a fast hardware/software co-simulation framework, is a key enabler to a fast embedded system design cycle. Unfortunately, the problem of deriving such estimates without a detailed implementation available is difficult.In this paper we describe two approaches to solve software cost and performance estimation problem, and how they are used in an embedded system design environment. A source-based approach uses compilation onto ...

49 Surfing the net for software engineering notes: Surfing the net for software engineering notes

Mark Doernhoefer
January 2005 **ACM SIGSOFT Software Engineering Notes**, Volume 30 Issue 1
Publisher: ACM Press
Full text available: pdf(1.75 MB) Additional Information: full citation, abstract, index terms

Building real-time applications can be one of the most difficult jobs facing today's software engineers. With all the talk about Web services and Web based applications we sometimes forget that there is a large community of practice engaged in the construction of hard real-time systems. Real-time software engineering has been around for several decades. Typical real-time applications can be found in the areas of communications, avionics, process control, and other specialized applications such a ...

50 Articles: Division of Labor in Embedded Systems

Ivan Goddard
April 2003 **Queue**, Volume 1 Issue 2
Publisher: ACM Press
Full text available: html(37.05 KB) Additional Information: full citation, index terms

51 RTK-Spec TRON: A Simulation Model of an ITRON Based RTOS Kernel in SystemC

M. AbdElSalam Hassan, Keishi Sakanushi, Yoshinori Takeuchi, Masaharu Imai
March 2005 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 1 DATE '05**
Publisher: IEEE Computer Society
Full text available: pdf(242.86 KB) Additional Information: full citation, abstract, citings, index terms

This paper presents the methodology and the modeling constructs we have developed to capture the real time aspects of RTOS simulation models in a System Level Design Language (SLDL) like SystemC. We describe these constructs and show how they are

used to build a simulation model of an RTOS kernel targeting the µ-ITRON OS specification standard.

52 The Next Bang: The Explosive Combination of Embedded Linux, XML and Instant Messaging
Doc Searls
September 2000 **Linux Journal**
Publisher: Specialized Systems Consultants, Inc.
Full text available: html(34.52 KB) Additional Information: full citation, references, index terms

53 Scalability, performance, and real-time: A programmable microkernel for real-time systems
Christoph M. Kirsch, Marco A. A. Sanvido, Thomas A. Henzinger
June 2005 **Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments VEE '05**
Publisher: ACM Press
Full text available: pdf(275.28 KB) Additional Information: full citation, abstract, references, citings, index terms

We present a new software system architecture for the implementation of hard real-time applications. The core of the system is a microkernel whose reactivity (interrupt handling as in synchronous reactive programs) and proactivity (task scheduling as in traditional RTOSs) are fully programmable. The microkernel, which we implemented on a StrongARM processor, consists of two interacting domain-specific virtual machines, a reactive E (Embedded) machine and a proactive S (Scheduling) machine. The m ...

Keywords: operating system, real time, virtual machine

54 Communication synthesis for distributed embedded systems
Ross B. Ortega, Gaetano Borriello
November 1998 **Proceedings of the 1998 IEEE/ACM international conference on Computer-aided design ICCAD '98**
Publisher: ACM Press
Full text available: pdf(996.36 KB) Additional Information: full citation, references, citings, index terms

Keywords: bus protocols, communication synthesis, distributed heterogeneous embedded systems, hardware/software co-synthesis, interprocessor communication, multihop communication

55 PACT 2001 workshops: Energy characterization of embedded real-time operating systems
Andrea Acquaviva, Luca Benini, Bruno Riccó
December 2001 **ACM SIGARCH Computer Architecture News**, Volume 29 Issue 5
Publisher: ACM Press
Full text available: pdf(484.99 KB) Additional Information: full citation, abstract, references, index terms

In this paper we propose a methodology to analyze the energy overhead due to the presence of an embedded operating system in a wearable device. Our objective is to determine the key parameters affecting the energy consumption of the RTOS allowing the development of more efficient OS-based power management policies. To achieve this target, we propose a characterization strategy that stimulates the RTOS both at the kernel and at the I/O driver level by analyzing various OS-related parameters. Our ...

56 Formal Methods for Integration of Automotive Software
Marek Jersak, Kai Richter, Rolf Ernst, Jorn-Christian Braam, Zheng-Yu Jiang, Fabian Wolf
March 2003 **Proceedings of the conference on Design, Automation and Test in Europe: Designers' Forum - Volume 2 DATE '03**

Publisher: IEEE Computer Society

Full text available: pdf(136.13 KB) Publisher Site

Additional Information: full citation, abstract, citings, index terms

Novel functionality, configurability and higher efficiency in automotive systems require sophisticated embedded software, as well as distributed software development between manufacturers and control unit suppliers. However, at least for engine control units, there exists today no well-defined software integration process that satisfies all key requirements of automotive manufacturers. We propose a methodology for safe integration of automotive software functions where required performance infor ...

57 Design space exploration and architectural design of HW/SW systems: Hardware support for real-time embedded multiprocessor system-on-a-chip memory management

Mohamed Shalan, Vincent J. Mooney

May 2002 **Proceedings of the tenth international symposium on Hardware/software codesign CODES '02**

Publisher: ACM Press

Full text available: pdf(533.74 KB) Additional Information: full citation, abstract, references, citings, index terms, review

The aggressive evolution of the semiconductor industry --- smaller process geometries, higher densities, and greater chip complexity --- has provided design engineers the means to create complex high-performance Systems-on-a-Chip (SoC) designs. Such SoC designs typically have more than one processor and huge memory, all on the same chip. Dealing with the global on- chip memory allocation/de-allocation in a dynamic yet deterministic way is an important issue for the upcoming billion transistor mu ...

Keywords: Atalanta, SoCDMMU, System-on-a-Chip, dynamic memory management, embedded systems, real-time operating systems., real-time systems, two-level memory management

58 Timed executable system specification of an ADSL modem using a C++ based design environment: a case study

Dirk Desmet, Michiel Esvelt, Probhat Avasare, Diederik Verkest, Hugo De Man

March 1999 **Proceedings of the seventh international workshop on Hardware/software codesign CODES '99**

Publisher: ACM Press

Full text available: pdf(437.81 KB) Additional Information: full citation, references, citings, index terms

59 Design and Implementation of embedded software: A memory-optimal buffering protocol for preservation of synchronous semantics under preemptive scheduling

Christos Sofronis, Stavros Tripakis, Paul Caspi

October 2006 **Proceedings of the 6th ACM & IEEE International conference on Embedded software EMSOFT '06**

Publisher: ACM Press

Full text available: pdf(341.34 KB) Additional Information: full citation, abstract, references, index terms

Recently, we have proposed a set of buffering schemes to preserve the semantics of a synchronous program when the latter is implemented as a set of multiple tasks running under preemptive scheduling. These schemes, however, are not optimal in terms of memory (buffer usage). In this paper we propose a new protocol which generalizes the previous schemes. The new protocol is not only semantics-preserving but also memory-optimal in two senses: first, in terms of the number of buffers required to pre ...

Keywords: embedded software, model-based design, optimality, preemptive scheduling, process communication, semantical preservation, synchronous programming

60 ECL: a specification environment for system-level design

Luciano Lavagno, Ellen Sentovich

June 1999 **Proceedings of the 36th ACM/IEEE conference on Design automation DAC '99**

Publisher: ACM Press

Full text available: pdf(82.31 KB) Additional Information: full citation, references, citings, index terms

Terms used **embedded rtos source code generation** Found **105,379** of **198,310**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 61 - 80 of 200 Result page: previous 1 2 3 **4** 5 6 7 8 9 10 next

Best 200 shown Relevance scale

61 Experience with an embedded systems software course

Jogesh K. Muppala

October 2005 **ACM SIGBED Review**, Volume 2 Issue 4

Publisher: ACM Press

Full text available: pdf(384.00 KB) Additional Information: full citation, abstract, references, index terms

In this paper we share our experience with designing and offering a senior undergraduate course on Embedded Systems Software in the Department of Computer Science at the Hong Kong University of Science and Technology. We give a detailed overview of the course, some reflections on our experience with the first offering of the course, followed by some discussion on the students' views of the course.

62 The embedded software consortium of taiwan

Tai-Yi Huang, Chung-Ta King, Youn-Long Steve Lin, Yin-Tsung Hwang

August 2005 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 4 Issue 3

Publisher: ACM Press

Full text available: pdf(422.96 KB) Additional Information: full citation, abstract, references, index terms

The advancement of semiconductor manufacturing technology makes it practical to place a traditional board-level embedded system on a single chip. The evolvement of system-on-chip (SoC) techniques presents new challenges for integrated circuit designs as well as embedded software and systems. To address these challenges, the Ministry of Education (MOE) of Taiwan has been running the VLSI Circuits and Systems Education Program since 1996. This program adopts a top-down approach by forming six doma ...

Keywords: Embedded software, educational curricula, integrated circuit design

63 Active base stations and nodes for wireless networks

Athanassios Boulis, Paul Lettieri, Mani Srivastava

January 2003 **Wireless Networks**, Volume 9 Issue 1

Publisher: Kluwer Academic Publishers

Full text available: pdf(441.19 KB) Additional Information: full citation, abstract, references, citings, index terms

Mobile and wireless network systems are characterized by a highly time varying and heterogeneous operational environment. For example, the wireless link bandwidth and bit error rate can change due to fading, mobile nodes may have different capabilities, and in the course of its movements a mobile node may visit base stations that provide different sets of services, protocols, and interfaces. Adaptability, in various forms and at various levels of the system, is a key to combating the inherent va ...

Keywords: active networking, base station, reconfigurable hardware, wireless and mobile nodes

64 A tool for performance estimation of networked embedded end-systems

Asawaree Kalavade, Pratyush Moghé

May 1998 **Proceedings of the 35th annual conference on Design automation DAC '98**

Publisher: ACM Press

Full text available: pdf(408.34 KB) Additional Information: full citation, abstract, references, citings, index terms

Networked embedded systems are expected to support adaptive streaming audio/video applications with soft real-time constraints. These systems can be designed in a cost efficient manner only if their architecture exploits the "leads" suggested by clever compile-time performance estimators. However, performance estimation of networked embedded systems is a non-trivial problem. The computational requirements of such systems show statistical variations that stem from several ...

Keywords: design methodology, microprocessor, timing, verification

65 Embedded tutorial: Code generation for embedded processors

Rainer Leupers

September 2000 **Proceedings of the 13th international symposium on System synthesis ISSS '00**

Publisher: IEEE Computer Society

Full text available: pdf(62.34 KB) Additional Information: full citation, abstract, references, citings

The increasing use of programmable processors as IP blocks in embedded system design creates a need for C/C++ compilers capable of generating efficient machine code. Many of today's compilers for embedded processors suffer from insufficient code quality in terms of code size and performance. This violates the tight chip area and real-time constraints often imposed on embedded systems. The reason is that embedded processors typically show architectural features which are not well handled by class ...

66 Efficient power co-estimation techniques for system-on-chip design

Marcello Lajolo, Anand Raghunathan, Sujit Dey

January 2000 **Proceedings of the conference on Design, automation and test in Europe DATE '00**

Publisher: ACM Press

Full text available: pdf(124.92 KB) Publisher Site Additional Information: full citation, references, citings, index terms

67 Specification and dynamic properties: Dynamic online reconfiguration for customizable and self-optimizing operating systems

Simon Oberthür, Carsten Böke, Björn Griese

September 2005 **Proceedings of the 5th ACM international conference on Embedded software EMSOFT '05**

Publisher: ACM Press

Full text available: pdf(268.46 KB) Additional Information: full citation, abstract, references, index terms

When applications adapt their behavior to the requirements of the environment, their resource usage can change dramatically. The resource usage implies the services that the applications require from the operating system. Thus, the operating system must either provide all services that are totally required over time or reconfigure itself. Reconfiguration of the operating system means to support on demand services or the possibility to degrade services. We present an approach where we extend our ...

Keywords: real-time operating system, self-optimizing

68 Modeling and control of robotic mine haulage system

Ernest G. Holzmann, Kenneth B. Haefner

March 1982 **Proceedings of the 15th annual symposium on Simulation ANSS '82**

Publisher: IEEE Computer Society Press
Full text available: pdf(1.03 MB) Additional Information: full citation, abstract, index terms

Concepts borrowed from discrete and hybrid system simulation have been successfully applied in the design of a complex, distributed computer control system. The example presented is a rail haulage system for a horizontally cut coal mine. The system controls all train movements and haulage operations into and out of the mine. The control logic is embedded in the system software, distributed over three levels: 1. Central control (resident in a minicomputer) 2. Wayside co ...

69 A compilation-based software estimation scheme for hardware/software co-simulation

Marcello Lajolo, Mihai Lazarescu, Alberto Sangiovanni-Vincentelli
March 1999 **Proceedings of the seventh international workshop on Hardware/software codesign CODES '99**
Publisher: ACM Press
Full text available: pdf(437.23 KB) Additional Information: full citation, references, citings, index terms

Keywords: compilation, delay modeling, software estimation

70 Co-design architecture and synthesis: Optimization and synthesis for complex reactive embedded systems by incremental collapsing

Massimiliano Chiodo
May 2002 **Proceedings of the tenth international symposium on Hardware/software codesign CODES '02**
Publisher: ACM Press
Full text available: pdf(500.99 KB) Additional Information: full citation, abstract, references, index terms

We propose a software synthesis procedure for reactive real-time embedded systems. In our approach, control parts of the system are represented in a decomposed form enabling more complex control structures to be represented. We propose a synthesis procedure for this representation that incrementally aggregates elements of the representation while keeping the resulting code size under tight control. This method combined with heuristic strategies works very well on real-life designs and demonstrat ...

Keywords: embedded systems, finite-state machines, real-time systems, software synthesis

71 Research trends in real-time computing for embedded systems

Giorgio Buttazzo
July 2006 **ACM SIGBED Review**, Volume 3 Issue 3
Publisher: ACM Press
Full text available: pdf(77.34 KB) Additional Information: full citation, abstract, references, index terms

Most of today's embedded systems are required to work in dynamic environments, where the characteristics of the computational load cannot always be predicted in advance. Still timely responses to events have to be provided within precise timing constraints in order to guarantee a desired level of performance. Hence, embedded systems are, by nature, inherently real-time. Moreover, most of embedded systems work under several resource constraints, due to space, weight, energy, and cost limitations ...

72 The CAN microcluster: Parallel processing over the controller area network

Paul A. Kuban, Rammohan K. Ragade
March 2005 **Journal on Educational Resources in Computing (JERIC)**, Volume 5 Issue 1
Publisher: ACM Press
Full text available: pdf(242.37 KB) Additional Information: full citation, abstract, references, index terms

Most electrical engineering and computer science undergraduate programs include at least one course on microcontrollers and assembly language programming. Some departments offer legacy courses in C programming, but few include C programming from an

embedded systems perspective, where it is still regularly used. Distributed computing and parallel processing are subjects generally reserved for graduate programs or specialized degrees. And although it is common to provide undergraduate courses on c ...

Keywords: CAN, Cluster, controller area network, distributed, embedded systems, microcontrollers, parallel

73 Architectural exploration and system simulations: Virtual synchronization technique with OS modeling for fast and time-accurate cosimulation

Youngmin Yi, Dohyung Kim, Soonhoi Ha

October 2003 **Proceedings of the 1st IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis CODES+ISSS '03**

Publisher: ACM Press

Full text available: pdf(279.75 KB) Additional Information: full citation, abstract, references, citings, index terms

Hardware/Software cosimulation is the key process to shorten the design turn around time. We have proposed a novel technique, called virtual synchronization, for fast and time accurate cosimulation that involves interacting component simulators. In this paper, we further extend the virtual synchronization technique with OS modeling for the case where multiple software tasks are executed under the supervision of a real-time operating system. The OS modeler models the RTOS overheads of context swi ...

Keywords: OS modeling, cosimulation, virtual synchronization

74 The HIJA project: 2: Issues in building an ANRTS platform

Antonio Kung, James Hunt, Ludovic Gauthier, Marc Richard-Foy

October 2006 **Proceedings of the 4th international workshop on Java technologies for real-time and embedded systems JTRES '06**

Publisher: ACM Press

Full text available: pdf(359.44 KB) Additional Information: full citation, abstract, references, index terms

The HIJA initiative is currently working on creating the technology conditions to achieve architecture neutrality for real-time systems. To this end it has developed a number of profiles based on RTSJ and developed a number of proofs of concept. In doing so, HIJA came across a number of integration issues which are not specific to the chosen technologies, and therefore need to be addressed by the embedded systems technology community in the large. This paper reports on the following issues: (1) ...

Keywords: architecture neutral platform, process integration, resource enforcement, space and time segregation

75 Mobile phones: the embedded linux challenge

Bill Weinberg

August 2006 **Linux Journal**, Volume 2006 Issue 148

Publisher: Specialized Systems Consultants, Inc.

Full text available: html(28.07 KB) Additional Information: full citation, abstract, index terms

Linux can be the ultimate embedded operating system if you know the secrets.

76 Session 16: special session: MPSOC design tools: Overview of the MPSoC design challenge

Grant Martin

July 2006 **Proceedings of the 43rd annual conference on Design automation DAC '06**

Publisher: ACM Press

Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, index terms

We review the design challenges faced by MPSoC designers at all levels. Starting at the application level, there is a need for programming models and communications APIs that allow applications to be easily re-configured for many different possible architectures without tedious rewriting, while at the same time ensuring efficient production code.

Synchronisation and control of task scheduling may be provided by RTOS's or other scheduling methods, and the choice of programming and threading model ...

Keywords: MPSoC, multi-processor system-on-chip, system-level design

77 Operating systems: *t-kernel*: providing reliable OS support to wireless sensor networks

Lin Gu, John A. Stankovic

October 2006 **Proceedings of the 4th international conference on Embedded networked sensor systems SenSys '06**

Publisher: ACM Press

Full text available: pdf(524.19 KB) Additional Information: full citation, abstract, references, index terms

The development of a reliable large-scale wireless sensor network (WSN) is very difficult because of resource constraints, energy budget, and demanding application requirements. Three OS features-OS protection, virtual memory, and preemptive scheduling-can significantly improve the reliability of WSN systems and facilitate developing complex WSN software. However, due to the lack of hardware support for privileged execution and address translation, it is impossible to implement these features wi ...

Keywords: OS protection, binary translation, low-power systems, virtual memory, wireless sensor networks

78 Real-time convergence of Ada and Java™

Ben Brosgol, Brian Dobbing

September 2001 **ACM SIGAda Ada Letters , Proceedings of the 2001 annual ACM SIGAda international conference on Ada SIGAda '01**, Volume XXI Issue 4

Publisher: ACM Press

Full text available: pdf(191.98 KB) Additional Information: full citation, abstract, references, citings, index terms

Two independent recent efforts have defined extensions to the Java platform that intend to satisfy real-time requirements. This paper summarizes the major features of these efforts, compares them to each other and to Ada 95's Real-Time Annex, and argues that their convergence with Ada95 may serve to complement rather than compete with Ada in the real-time domain.

Keywords: Ada, Java, Real-Time, asynchrony, garbage collection, scheduling, threads

79 Fast hardware-software co-simulation using VHDL models

Bassam Tabbara, Enrica Filippi, Luciano Lavagno

January 1999 **Proceedings of the conference on Design, automation and test in Europe DATE '99**

Publisher: ACM Press

Full text available: pdf(548.81 KB) Additional Information: full citation, citings, index terms

80 Correct-by-Construction Transformations across Design Environments for Model-Based Embedded Software Development

M. Baleani, A. Ferrari, L. Mangeruca, A. L. Sangiovanni-Vincentelli, U. Freund, E. Schlenker, H.-J. Wolff

March 2005 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 2 DATE '05**

Publisher: IEEE Computer Society

Full text available: pdf(471.02 KB) Additional Information: full citation, abstract, index terms

Embedded software design for real time reactive system has become the bottleneck in the market introduction of complex products such as automobiles, airplanes, and industrial control plants. In particular, functional correctness and reactive performance are increasingly difficult to verify. The advent of model-based design methodologies has

alleviated some of the verification-related problems by making the code-generation process flow automatically from the model description. Given the relative ...

Terms used **embedded rtos source code generation** Found **105,379** of **198,310**

Sort results by: relevance — Save results to a Binder — Search Tips — Open results in a new window

Display results: expanded form

Try an Advanced Search
Try this search in The ACM Guide

Results 81 - 100 of 200 Result page: previous 1 2 3 4 **5** 6 7 8 9 10 next

Best 200 shown Relevance scale

81 Experiences from large embedded systems development projects in education, involving industry and research

Martin Törngren, Martin Grimheden, Niklas Adamsson
January 2007 **ACM SIGBED Review**, Volume 4 Issue 1

Publisher: ACM Press

Full text available: pdf(501.65 KB) Additional Information: full citation, abstract, references, index terms

We present experiences from a final year M.Sc. course. The overall aim of the course is to provide knowledge and skills to develop products in small or large development teams. The course is implemented in terms of large projects in cooperation with external partners, in which the students, based on a product specification, apply and integrate their accumulated knowledge in the development of a prototype. This course, which has been running and further elaborated for 20 years, has been proven su ...

82 Embedding Linux to Control Accelerators and Experiments

A. Gotz, P. Makijarvi, B. Regad, M. Perez, P. Mangiagalli
October 1999 **Linux Journal**

Publisher: Specialized Systems Consultants, Inc.

Full text available: html(28.17 KB) Additional Information: full citation, abstract, references, index terms

A scientific laboratory in Europe depends on Linux for controlling equipment used in their research

83 On-chip communication and interface design: Automatic network generation for system-on-chip communication design

Dongwan Shin, Andreas Gerstlauer, Rainer Dömer, Daniel D. Gajski
September 2005 **Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis CODES+ISSS '05 , Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis CODES+ISSS '05**

Publisher: ACM Press, IEEE Computer Society

Full text available: pdf(206.97 KB) Publisher Site Additional Information: full citation, abstract, references, index terms

With growing system complexities, system-level communication design is becoming increasingly important and advanced, network-oriented communication architectures become necessary. In this paper, we extend previous work on automatic communication refinement to support non-traditional, network-oriented architectures beyond a single bus. From an abstract description of the desired communication channels, the refinement tools automatically generate executable models and implementations of the system ...

Keywords: communication synthesis, system level design

84 Embedded software: Embedded software generation from system level design languages

Haobo Yu, Rainer Dömer, Daniel Gajski

January 2004 **Proceedings of the 2004 conference on Asia South Pacific design automation: electronic design and solution fair ASP-DAC '04 , Proceedings of the 2004 conference on Asia South Pacific design automation: electronic design and solution fair ASP-DAC '04**

Publisher: IEEE Press

Full text available: pdf(108.36 KB) Publisher Site Additional Information: full citation, abstract, references, citings

To meet the challenge of increasing design complexity, designers are turning to system level design languages (SLDLs) to model systems at a higher level of abstraction. This paper presents a method of automatically generating embedded software from system specification written in SLDL. Several refinement steps and intermediate models are introduced in our software generation flow. We demonstrate the effectiveness of the proposed method by a tool which can generate efficient ANSI C code from syst ...

85 Focus on Embedded Systems

Rick Lehrbaum

November 2000 **Linux Journal**

Publisher: Specialized Systems Consultants, Inc.

Full text available: html(15.48 KB) Additional Information: full citation, index terms

86 Optimized rapid prototyping for real-time embedded heterogeneous multiprocessors

T. Grandpierre, C. Lavarenne, Y. Sorel

March 1999 **Proceedings of the seventh international workshop on Hardware/software codesign CODES '99**

Publisher: ACM Press

Full text available: pdf(515.87 KB) Additional Information: full citation, references, citings, index terms

87 Poster session I: Enabling RTOS simulation modeling in a system level design language

M. AbdElSalam Hassan, Keishi Sakanushi, Yoshinori Takeuchi, Masaharu Imai

January 2005 **Proceedings of the 2005 conference on Asia South Pacific design automation ASP-DAC '05**

Publisher: ACM Press

Full text available: pdf(346.94 KB) Additional Information: full citation, abstract, references

In this paper, we propose a new process definition (T-THREAD) and an extension to the existing SystemC simulation engine (SIM_API library) to capture the real time aspects of RTOS simulation models in an SLDL like SystemC. We describe the execution semantics of this process and show how it works in a complete embedded system simulation model.

88 On-chip communication and interface design: A multicast inter-task communication protocol for embedded multiprocessor systems

Víctor Reyes, Tomás Bautista, Gustavo Marrero, Antonio Núñez, Wido Kruijtzer

September 2005 **Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis CODES+ISSS '05 , Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis CODES+ISSS '05**

Publisher: ACM Press, IEEE Computer Society

Full text available: pdf(282.77 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Recently, a new programming model and platform interface for MPSoC design and integration called TTL (Task Transaction Level) has been developed and advocated as a standard. In this paper, a specific implementation of the TTL interface named ITCP (Inter-

Task Communication Protocol) is presented. ITCP is well suited for both hardware and software implementations and supports features such as multitasking and multicast communication. A configurable SystemC model of the ITCP protocol and its integr ...

Keywords: multiprocessor design, parallel programming model, platform interface, task transaction level

89 System level modelling and simulation: A SW performance estimation framework for early system-level-design using fine-grained instrumentation

Torsten Kempf, Kingshuk Karuri, Stefan Wallentowitz, Gerd Ascheid, Rainer Leupers, Heinrich Meyr

March 2006 **Proceedings of the conference on Design, automation and test in Europe: Proceedings DATE '06**

Publisher: European Design and Automation Association

Full text available: pdf(1.63 MB) Additional Information: full citation, abstract, references

The increasing demands of high-performance in embedded applications under shortening time-to-market has prompted system architects in recent time to opt for Multi-Processor Systems-on-Chip (MP-SoCs) employing several programmable devices. The programmable cores provide a high amount of flexibility and reusability, and can be optimized to the requirements of the application to deliver high-performance as well. Since application software forms the basis of such designs, the need to tune the underl ...

90 A dynamic memory management unit for embedded real-time system-on-a-chip

Mohamed Shalan, Vincent J. Mooney

November 2000 **Proceedings of the 2000 international conference on Compilers, architecture, and synthesis for embedded systems CASES '00**

Publisher: ACM Press

Full text available: pdf(321.80 KB) Additional Information: full citation, citings

Keywords: SoCDMMU, dynamic memory management, embedded systems, real-time systems, system-on-a-chip, two-level memory management

91 Efficient code generation from SHIM models

Stephen A. Edwards, Olivier Tardieu

June 2006 **ACM SIGPLAN Notices , Proceedings of the 2006 ACM SIGPLAN/SIGBED conference on Language, compilers and tool support for embedded systems LCTES '06**, Volume 41 Issue 7

Publisher: ACM Press

Full text available: pdf(128.41 KB) Additional Information: full citation, abstract, references, index terms

Programming concurrent systems is substantially more difficult than programming sequential systems, yet most embedded systems need concurrency. We believe this should be addressed through higher-level models of concurrency that eliminate many of the usual challenges, such as nondeterminism arising from races.The shim model of computation provides deterministic concurrency, and there already exist ways of implementing it in hardware and software. In this work, we describe how to produce ...

Keywords: code synthesis, computed gotos, concurrency, embedded systems, the shim model

92 Formal synthesis and code generation of embedded real-time software

Pao-Ann Hsiung

April 2001 **Proceedings of the ninth international symposium on Hardware/software codesign CODES '01**

Publisher: ACM Press

Full text available: pdf(456.04 KB) Additional Information: full citation, abstract, references, citings, index terms

Due to rapidly increasing system complexity, shortening time-to-market, and growing demand for hard real-time systems, formal methods are becoming indispensable in the synthesis of embedded systems, which must satisfy stringent temporal, memory, and environment constraints. There is a general lack of practical formal methods that can synthesize complex embedded real-time software (ERTS). In this work, a formal method based on *Time Free-Choice Petri Nets* (TFCPN) is proposed for ERTS syn ...

Keywords: Petri Nets, code generation, embedded real-time software, scheduling

93 Poster session 2: Hardware/software partitioning of operating systems: a behavioral synthesis approach

Sathish Chandra, Francesco Regazzoni, Marcello Lajolo

April 2006 **Proceedings of the 16th ACM Great Lakes symposium on VLSI GLSVLSI '06**

Publisher: ACM Press

Full text available: pdf(387.60 KB) Additional Information: full citation, abstract, references, index terms

In this paper we propose a hardware real time operating system(HW-RTOS) solution that makes use of a dedicated hardware in order to replace the standard support provided by the POSIX layer of a general purpose RTOS for implementing task synchronization and scheduling. By redefining only the I/O APIs of the tasks, the HW-RTOS then takes care of the communication requirements of the original application and also implements the task scheduling algorithm. The new software application can then be com ...

Keywords: HW/SW codesign, RTOS, embedded systems, operating systems, partitioning

94 OOPSLA practitioner reports chair's welcome: Using model-driven engineering to complement software product line engineering in developing software defined radio components and applications

Bruce Trask, Dominick Paniscotti, Angel Roman, Vikram Bhanot

October 2006 **Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06**

Publisher: ACM Press

Full text available: pdf(1.14 MB) Additional Information: full citation, abstract, references, index terms

This paper details the application of Software Product Lines (SPL)[16] and Model-Driven Engineering (MDE)[15] to the software defined radio domain. More specifically it is an experience report emphasizing the synergy [17] resulting from combining MDE and SPL technologies. The software defined radio domain has very unique characteristics as its systems typically are a confluence of a number of typically challenging aspects of software development. To name a few, these s ...

Keywords: development, domain, generation, language, model

95 Composable code generation for distributed giotto

Thomas A. Henzinger, Christoph M. Kirsch, Slobodan Matic

June 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools for embedded systems LCTES '05**, Volume 40 Issue 7

Publisher: ACM Press

Full text available: pdf(294.16 KB) Additional Information: full citation, abstract, references, index terms

We present a compositional approach to the implementation of hard real-time software running on a distributed platform. We explain how several code suppliers, coordinated by a system integrator, can independently generate different parts of the distributed software. The task structure, interaction, and timing is specified as a Giotto program. Each supplier is given a part of the Giotto program and a timing interface, from which the supplier generates task and scheduling code. The integrator then ...

Keywords: distributed compilation, real time

96 Unified Component Integration Flow for Multi-Processor SoC Design and Validation
Mohamed-Anouar Dziri, W. Cesário, Flávio R. Wagner, A. A. Jerraya
February 2004 **Proceedings of the conference on Design, automation and test in Europe - Volume 2 DATE '04**
Publisher: IEEE Computer Society
Full text available: pdf(179.09 KB) Additional Information: full citation, abstract, citings, index terms

Most system-on-Chip (SoC) design methodologies promote the reuse of pre-designed (hardware, software, and functional) components. However, as these components are heterogeneous, their integration requires complex interface sub-systems. These sub-systems can also be constructed by assembling pre-designed basic interface components. Hence, SoC design and validation involves component composition techniques to create hardware, software, and functional interface sub-systems by assembling basic inter ...

97 SPOTS'06 session 4--new sensors and architectures: The low power energy aware processing (LEAP)embedded networked sensor system
Dustin McIntire, Kei Ho, Bernie Yip, Amarjeet Singh, Winston Wu, William J. Kaiser
April 2006 **Proceedings of the fifth international conference on Information processing in sensor networks IPSN '06**
Publisher: ACM Press
Full text available: pdf(200.80 KB) Additional Information: full citation, abstract, references, index terms

A broad range of embedded networked sensor (ENS) systems for critical environmental monitoring applications now require complex, high peak power dissipating sensor devices, as well as on-demand high performance computing and high bandwidth communication. Embedded computing demands for these new platforms include support for computationally intensive image and signal processing as well as optimization and statistical computing. To meet these new requirements while maintaining critical support for ...

Keywords: embedded wireless networked sensor, energy-aware multiprocessor platform, sensor platform hardware and software architecture

98 A mapping algorithm for computer-assisted exploration in the design of embedded systems
E. P. Mariatos, A. N. Birbas, M. K. Birbas
January 2001 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 6 Issue 1
Publisher: ACM Press
Full text available: pdf(296.29 KB) Additional Information: full citation, abstract, references, index terms, review

We present a technique for automatic exploration of architectural alternatives in the design of complex electronic embedded systems and systems-on-a-chip. The technique transforms the problem into a set of simple model-to-model operations and a mapping algorithm that becomes the core of the entire design process. The mapping algorithm is formulated as an assignment-type problem (ATP), which is, in turn, solved by a straightforward optimization method. The result is a design assistance tool, ...

Keywords: codesign, embedded system design space exploration, specification mapping

99 Graph-based code selection techniques for embedded processors
October 2000 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 5 Issue 4
Publisher: ACM Press
Full text available: pdf(356.83 KB) Additional Information: full citation, abstract, references, index terms, review

Code selection is an important task in code generation for programmable processors, where the goal is to find an efficient mapping of machine-independent intermediate code to processor-specific machine instructions. Traditional approaches to code selection are

based on tree parsing which enables fast and optimal code selection for intermediate code given as a set of data-flow trees. While this approach is generally useful in compilers for general-purpose processors, it may lead to poor code ...

Keywords: SIMD instructions, code selection, data-flow graphs, embedded processors, irregular data paths

100 Transaction-level modeling and exploration: Automatic generation of transaction level models for rapid design space exploration

Dongwan Shin, Andreas Gerstlauer, Junyu Peng, Rainer Dömer, Daniel D. Gajski

October 2006 **Proceedings of the 4th international conference on Hardware/software codesign and system synthesis CODES+ISSS '06**

Publisher: ACM Press

Full text available: pdf(192.99 KB) Additional Information: full citation, abstract, references, index terms

Transaction-level modeling has been touted to improve simulation performance and modeling efficiency for early design space exploration. But no tools are available to generate such transaction-level models from abstract input descriptions. Designers have to write such models manually, which is a tedious and error-prone task, and one of bottlenecks in improving designer's productivity. In this paper, we propose a method to generate transaction-level models from virtual architecture models where c ...

Keywords: communication synthesis, transaction-level model